CLAIMS

1. A method comprising:

assigning each of a plurality of segments comprising a received corpus to a node in a data structure denoting dependencies between nodes; and

calculating a transitional probability between each of the nodes in the data structure.

2. A method according to claim 1, further comprising:

calculating a frequency for each elemental item of the segment; and

removing nodes of the data structure associated with items which do not meet a minimum frequency threshold.

3. A method according to claim 2, wherein the frequency of the item is calculated by counting item occurrences throughout the subset and/or corpus.

4. A method according to claim 2, wherein the minimum threshold is three (3).

5. A method according to claim 1, further comprising:

managing storage of the data structure across a system memory of a computer system and an extended memory of the computer system.

6. A method according to claim 5, wherein the step of managing storage of the data structure comprises:

identifying least recently used nodes of the data structure; and

storing the least recently used nodes of the data structure in the extended memory of the computer system when the data structure is too large to store completely within the system memory.

7. A method according to claim 5, wherein the extended memory of the computer system comprises one or more files on an accessible mass storage device.

8. A method according to claim 7, wherein the data structure represents a language model, spread across one or more elements of a computing system memory subsystem.

9. A method according to claim 1, wherein calculating a transition probability includes calculating a Markov transitional probability between nodes.

10. A storage medium comprising a plurality of executable instructions including at least a subset of which that, when executed by a processor, implement a method according to claim 1.

11. A method for predicting a likelihood of an item in a corpus comprised of a plurality of items, the method comprising:

building a data structure of corpus segments representing a dynamic context of item dependencies within the segments;

calculating the likelihood of each item based, at least in part, on a likelihood of preceding items within the dynamic context; and

iteratively re-segmenting the corpus to improve the calculated likelihood of item dependencies.

12. A method according to claim 11, wherein the method of building a dynamic context of preceding dependent items comprises:

analyzing the data structure representing the language model;

identifying all items with dependencies to or from the item; and

using all items with dependencies to or from the item as the dynamic context.

13. A method according to claim 11, wherein the language model includes frequency information for each item within the model.

14. A method according to claim 13, wherein calculating the likelihood of the item comprises:

calculating a Markov transition probability for the item based, at least in part, on the frequency of the items comprising the dynamic context.

15. A method according to claim 11, wherein calculating the likelihood of the item comprises:

calculating a Markov transition probability for the item given the dynamic context of items.

16. A storage medium having stored thereon a plurality of executable instructions including instructions which, when executed by a host computer, implement a method according to claim 11.

17. A data structure, generated by a computer system as a statistical language model, the data structure comprising:

one or more root nodes; and

a plurality of subordinate nodes, ultimately linked to a root node, cumulatively comprising one or more sub-trees, wherein each node of a sub-tree represents one or more items of a corpus and includes a measure of a Markov transition probability between the node and another linked node.

18. A data structure according to claim 17, wherein the root node represents a common root item for all subordinate nodes in the one or more sub-trees.

19. A data structure according to claim 17, wherein the Markov transition probability is a measure of the likelihood of a transition from one node to another node based, at least in part, on the one or more items represented by each of the nodes.

20. A data structure according to claim 17, wherein the items include one or more of a character, a letter, a number, and combinations thereof.

21. A data structure according to claim 17, wherein the data structure represents a dynamic order Markov model (DOMM) language model of the textual source.

22. A storage medium comprising a plurality of executable instructions which, when executed by a processor, implement a data structure according to claim 17.

23. A memory subsystem in a computer system including one or more of a cache memory, a system memory and extended memory having information stored therein which, when interpreted by a processor of the computer system, represent a data structure according to claim 17.

24. A modeling agent comprising:

a controller, to receive a corpus; and

a data structure generator, responsive to and selectively invoked by the controller, to assign each of a plurality of segments comprising the received corpus to a node in a data structure denoting dependencies between nodes;

wherein the modeling agent calculates a transitional probability between each of the nodes of the data structure to determine a predictive capability of a language model represented by the data structure and iteratively re-segments the received corpus until a threshold predictive capability is reached.

25. A modeling agent according to claim 24, the data structure generator comprising:

a dynamic segmentation function, to iteratively re-segment the received corpus to improve language model predictive capability.

26. A modeling agent according to claim 24, the data structure generator comprising:

a frequency analysis function, to analyze a frequency of occurrence of segments within the corpus.

27. A modeling agent according to claim 26, wherein segments that do not meet a frequency of occurrence threshold are removed from the data structure, reducing data structure size and improving language model predictive capability.

28. A storage medium comprising a plurality of executable instructions including at least a subset of which, when executed, implement a language modeling agent to assign each of a plurality of segments of a received corpus to a node in a data structure denoting dependencies between nodes, and to calculate a transitional probability between each of the nodes in the data structure to determine a predictive capability of a language model denoted by the data structure, wherein the modeling agent dynamically re-segments the received corpus to remove segments which do not meet a minimum frequency threshold to improve one or more language model performance attributes.

29. A storage medium according to claim 28, wherein the one or more language model performance attributes include a predictive capability.